Online ISSN 1848-3380, Print ISSN 0005-1144
ATKAFF 57(4), 996–1006(2016)

Ramazan Havangi

Improved FastSLAM2.0 using ANFIS and PSO

DOI 10.7305/automatika.2017.12.1627
UDK 004.896.021.032.26:[159.937.52+528.023]

Original scientific paper

FastSLAM2.0 is a framework for simultaneous localization of robot using a Rao-Blackwellized particle filter (RBPF). One of the problems of FastSLAM2.0 relates to the design of RBPF. The performance and quality of the estimation of RBPF depends heavily on the correct a priori knowledge of the process and measurement noise covariance matrices that are in most real-life applications unknown. On the other hand, an incorrect a priori knowledge may seriously degrade their performance. This paper presents an intelligent RBPF to solve this problem. In this method, two adaptive Neuro-Fuzzy inference systems (ANFIS) are used for tuning the process and measurement noise covariance matrices and for increasing acuuracy and consistency. In addition, we use particle swarm optimization (PSO) to optimize the performance of sampling. Experimental results demonstrate that the proposed algorithm is effective.

Key words: simultaneous localization and mapping (SLAM), FastSLAM, ANFIS, PSO

Poboljšani FastSLAM2.0 algoritam korištenjem ANFIS-a i PSO-a. FastSLAM2.0 je algoritam za istodobnu lokalizaciju robota i kartiranje prostora koji koristi Rao-Blackwell verziju čestičnog filtra (RBPF). Jedan od problema FastSLAM2.0 algoritma je u dizajnu samog RBPF-a. Performanse i kvaliteta estimacije RBPF-a značajno ovisi o apriori poznavanju procesa i matrica kovarijanci mjernog šuma koje su za većinu procesa iz stvarnog svijeta nepoznate. S druge strane pogrešno pretpostavka može značajno narušiti performanse. Ovaj rad predstavlja inteligentnu verziju RBPF-a koja rješava ovaj problem. Predstavljena metoda koristi dva adaptivna neizrazito-neuronska sustava (ANFIS) za podešavanje matrica kovarijanci procesnog i mjernog šuma čime se povećava točnost i konzistencija RBPF algoritma. Također koristi se i optimizacija roja čestica (PSO) za optimiziranje performansi otipkavanja. Eksperimentalni rezultati pokazuju efikasnost predloženog algoritma.

Ključne riječi: istovremena lokalizacija i kartiranje (SLAM), FastSLAM, ANFIS, PSO

1 INTRODUCTION

The simultaneous localization and mapping (SLAM) is a fundamental problem of robots to perform autonomous tasks such as exploration in an unknown environment. It represents an important role in the autonomy of a mobile robot.

The most classical SLAM solutions are methods based on extended Kalman filter (EKF-SLAM) and based on Rao-Blackwellized particle filter (FastSLAM). However, EKF-SLAM suffers from two major problems: the computational complexity and data association [1-2]. Recently, the FastSLAM algorithm approach has been proposed as an alternative approach to solve the SLAM problem [3-5].

FastSLAM is an instance of Rao-Blackwellized particle filter, which partitions the SLAM posterior into a localization problem and an independent landmark position estimation problem. There are two versions of FastSLAM in the literature, FastSLAM1.0 and FastSLAM2.0 [4]. As FasSLAM2.0 is superior to FastSLAM1.0, we focus on the second version that for easily called FastSLAM in this paper. In FastSLAM, extended Kalman particle filter (EKPF) is used for the mobile robot position estimation and EKF is used for the feature location's estimation.

The key feature of FastSLAM is the fact that the data association decisions can be determined on a per-particle basis, and hence different particles can be associated with different landmarks. Each particle in FastSLAM may even have a different number of landmarks in its respective map. This characteristic gives FastSLAM the possibility of dealing with multi-hypothesis association problem. The ability to simultaneously pursue multiple data associations makes FastSLAM significantly more robust to the data association problems than other algorithms such as EKF-SLAM. The other advantage of FastSLAM arises from the fact that particle filters can cope with nonlinear and non-Gaussian robot motion models. There have been many investigations on FastSLAM. However, FastSLAM also has some draw-

backs. In [8-10], it has been noted that FastSLAM degenerates over time. This degeneracy is due to the fact that a particle set estimating the pose of the robot loses its diversity. One of the main reasons for loosing particle diversity in FastSLAM is sample impoverishment [8-10]. It occurs when likelihood lies in the tail of the proposal distribution [4]. Researchers have been trying to solve these problems in [4], [11-16]. In [16], a modified FastSLAM1.0 is presented by soft computing. In this algorithm, an adaptive neuro-fuzzy extended Kalman filter is used for landmark feature estimation and a novel multi swarm particle filter is presented to overcome the impoverishment of FastSLAM.

In all the aforementioned studies, however, FastSLAM has some drawbacks. A significant difficulty in designing of RBPF can often be traced to incomplete a priori knowledge of the process covariance matrix Q_t and measurement noise covariance matrix R_t. In most real applications, these matrixes are unknown. On the other hand, an incorrect a prior knowledge of Q_t and R_t may seriously degrade the RBPF performance. In this paper to solve these problems, an intelligent RBPF based SLAM is proposed. In this algorithm, two adaptive Neuro-Fuzzy inference systems(ANFIS) are used for tuning the process and measurement noise covariance matrices for increasing consistency. The free parameters of adaptive Neuro-Fuzzy inference systems are trained using the steepest gradient descent (GD) to minimize the differences of the actual value of the covariance of the residual with its theoretical value as much as possible. In addition, we use PSO to optimize the performance sampling of the RBPF. The PSO causes the particle set tend to the high likelihood region before the weight is updated, thereby the impoverishment of particles can be overcome.

The rest of the paper is organized as follows. In Section 2, FastSLAM problem is reviewed. The proposed algorithm is presented in Section 3. In Section 4, the simulation and experimental results are presented.

2 BACKGROUND: FASTSLAM

To describe SLAM, let us denote the map byΘ and the pose of the robot at time $t$bys_t. The map consists of a collection of features, each of which will be denoted byθ_n and the total number of stationary features will be denoted by$N$. In this situation, the SLAM problem can be formulized in a Bayesian probabilistic framework by representing each of the robot's position and map location as a probabilistic density function as [4]:

$$p(s_t, \Theta | z^t, u^t, n^t) \tag{1}$$

In essence, it is necessary to estimate the posterior density of maps Θ and poses s_t given that we know the observation$z^t = \{z_1, ..., z_t\}$, the control input $u^t = \{u_1, ..., u_t\}$ and the data associationn^t. Here, data association represents the mapping between map points in Θ and observation in z^t .The structure of SLAM enables particle filters to be applicable. This special particle filter is known as the Rao-Blackwellized particle filter (RBPF). The RBPF is introduced as an effective means to solve the SLAM problem. The term 'Rao-Blackwellized' means factoring of a state into a sampled part and an analytical part. The FastSLAM computes the posterior over maps and a robot's path. The key mathematical insight of FastSLAM pertains to the fact that the full SLAM posterior can be factorized as follows when data associationn^t is known [4]:

$$\begin{aligned} p(s^t, \Theta | z^t, u^t, n^t) &= p(s^t | z^t, u^t, n^t) \\ &\cdot \prod_{n=1}^{N} p(\theta_n | s^t, z^t, u^t, n^t) \end{aligned} \tag{2}$$

where $s^t = \{s_1, ..., s_t\}$ is a robot path or trajectory. This factorization states that the SLAM problem can be decomposed into estimating the product of a posterior over robot path and N landmark posteriors given the knowledge of the robot's path. The FastSLAM algorithm implements the path estimator $p(s^t | z^t, u^t, n^t)$ using a particle filter and the landmarks pose $p(\theta_n | s^t, z^t, u^t, n^t)$are realized by EKF, using separate filters for different landmarks. Each particle forms as follows [3-5]:

$$S_t^{[m]} =< s^{t,[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, ..., \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} > \tag{3}$$

where $[m]$ indicates the index of the particle, and $s^{t,[m]}$ is the $m-th$ particle's path estimate, and $\mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]}$ are the mean and the covariance of the Gaussian distribution representing the $n-th$ feature location conditioned on the path $s^{t,[m]}$. In general, it is not possible to draw samples directly from the SLAM posterior. Instead, the samples are drawn from a simpler distribution called the proposal distribution$q(s^{t,[m]} | z^t, u^t, n^t)$. The update algorithm of posterior FastSLAM consists of sampling, landmark update and resampling.

2.1 Sampling

In FastSLAM, the robot pose is sampled with respect to both the motion u_t and the measurement z_t as follows:

$$q(s_t^{[m]} | s^{t-1,[m]}, z^t, u^t, n^t) \tag{4}$$

An effective approach to accomplish this is to use EKF generated Gaussian approximation as follows:

$$q(s_t^{[m]} | s^{t-1,[m]}, z^t, u^t, n^t) \overset{EKF}{\sim} N(s_t, s_t^{[m]}, P_t^{[m]}) \tag{5}$$

For this purpose, each particle is predicted using EKF according to the following equations:

$$\hat{s}_{t+1}^{[m]} = f(s_t^{[m]}, u_t) \tag{6}$$

$$P_{t+1}^{[m]-} = \nabla f_t P_t^{[m]} \nabla f_t^T + G_u Q_t G_u^T \tag{7}$$

where

$$\nabla f_t = \frac{\partial f}{\partial s_t} \quad , \quad G_u = \frac{\partial f}{\partial u} \tag{8}$$

Then, the mean and covariance of the proposal distribution are updated at the measurement time by following equations:

$$s_t^{[m]} = \hat{s}_t^{[m]} + K_t^{[m]}(z_k - h(\hat{s}_t^{[m]})) \tag{9}$$

$$P_t^{[m]} = P_t^{[m]-} - P_t^{[m]-} H_t^T (S_t^{[m]})^{-1} H_t P_t^{[m]-} \tag{10}$$

where

$$\begin{aligned} K_t^{[m]} &= P_t^{[m]-} H_t^T (S_t^{[m]})^{-1} \\ S_t^{[m]} &= R + H_t P_t^{[m]-} H_t^T + G_{\theta_{n_t}} \Sigma_{n_t,t-1}^{[m]} G_{\theta_{n_t}}^T \end{aligned} \tag{11}$$

In above equations$\Sigma_{n_t,t-1}^{[m]}$ is the covariance of the landmark observed at time t but registered previously, H_t and $G_{\theta_{n_t}}$ are Jacobians. From the Gaussian distribution generated by the estimated mean and covariance of the vehicle, the state of each particle is sampled:

$$s_t^{[m]} \sim N(s_t^{[m]}, P_t^{[m]}) \tag{12}$$

When there is no observation, the vehicle state is predicted without the measurement update using (6) and (7). If many landmarks are observed at the same time, (9) and (10) are repeated for each observed landmark, and the mean and the covariance of the vehicle are updated based on the previously updated one.

2.2 Landmark Update

The FastSLAM algorithm represents the posterior landmark estimates $p(\theta_n | s^t, z^t, u^t, n^t)$ using low-dimensional EKFs. In fact FastSLAM updates the posterior over the landmark estimates, respected by the mean $\mu_{n,t-1}^{[m]}$ and the covariance $\Sigma_{n,t-1}^{[m]}$. The update of posterior depends on whether or not a landmark n was observed at time t. For $n \neq n_t$, the posterior over the landmark remains unchanged. For the observed feature$n = n_t$, the mean and covariance is as follows [4]:

$$\hat{z}_t = g(s_t^{[m]}, \mu_{n_t,t-1}) \tag{13}$$

$$G_{\theta_{n_t}} = \nabla_{\theta_{n_t}} g(s_t, \theta_{n_t})\big|_{s_t = s_t^{[m]}; \theta_{n_t} = \mu_{n_t,t-1}^{[m]}} \tag{14}$$

$$Z_{n,t} = G_{\theta_{n_t}} \Sigma_{n_t,t-1}^{[m]} G_{\theta_{n_t}}^T + R_t \tag{15}$$

$$K_t = \Sigma_{n_t,t-1}^{[m]} G_{\theta_{n_t}}^T Z_{n,t}^{-1} \tag{16}$$

$$\mu_{n_t,t}^{[m]} = \mu_{n_t,t-1}^{[m]} + K_t(z_t - \hat{z}_t) \tag{17}$$

$$\Sigma_{n_t,t}^{[m]} = (I - K_t G_{\theta_{n_t}}) \Sigma_{n_t,t-1}^{[m]} \tag{18}$$

2.3 Resampling

The importance weight of particles is computed by considering the most recent observation as follows:

$$w_t^{[m]} = w_{t-1}^{[m]} \frac{p(z_t | s_t^{[m]}) p(s_t^{[m]} | u^t, n^t)}{N(s_t, s_t^{[m]}, P_t^{[m]})} \tag{19}$$

Since the variance of the importance weights increase over time [3-5], resampling is required. In the resampling process, particles with low importance weight are eliminated and particles with high weights are multiplied.

3 IMPROVED FASTSLAM

As already motioned, FastSLAM uses a conventional RBPF. In RBPF, complete a priori knowledge of the process and measurement noise statistics is assumed. However, in real-life applications these matrices are unknown.

An incorrect a priori knowledge of Q_t andR_t may lead to performance degradation and inconsistency [17-18]. It can even lead to practical divergence [17-18].This is because EKF is used in the design of the proposal distribution and landmark position estimation. On the other hand, the performance of EKF depends largely on the accuracy of the knowledge of process covariance matrix Q_t and measurement noise covarianceR_t.

One of the efficient ways to overcome the above weakness is to use an adaptive algorithm. Two major approaches that have been proposed for adaptive EKF are multiple model adaptive estimation (MMAE) and innovation adaptive estimation (IAE) [17-18]. Although the implementation of these approaches is different, they both share the same concept of utilizing new statistical information obtained from the residual (innovation) sequence. In this paper, we adaptively tuned matrices Q_t and R_t by two ANFIS. This is because there are two EKF in FastSLAM, one is for updating the proposal distribution (EKPF) and other is for updating landmark's position estimation. As landmarks are static, R_t is adaptively tuned when the landmark position is updated. In addition, as measurements in EKF and EKPF are the same and R_t is tuned when the landmark position is estimated, the only Q_t is adaptively tuned when the proposal distribution is updated.

In proposed method, proposal distribution is generated by adaptive Neuro-Fuzzy extend Kalman particle filter to obtain a better importance sampling distribution. However, as the EKF approximates the nonlinear function in the state dynamic and measurement models, the degradation of the particles is still inevitable. To solve this problem and improve of samping, PSO [20-21] is merged in Neuro-Fuzzy extend Kalman particle filter before sampling to move particles towards region of the state space where a posterior probability density is significant. It can be overcome the

impoverishment of particles. While, the optimization process makes the particles which are far away from the true state tend to the region where the true state has a greater probability of emergence, it can enhance the effect of each particle.

The proposed algorithm is similar to FastSLAM and consists of sampling, feature update and calculating importance weight and resampling. The main difference between the proposed algorithm and FastSLAM is as follows:

- Tuning statistics
- Modifying of sampling

3.1 Tuning statistics

a) Tuning R_t

The covariance matrix R_t represents the accuracy of the measurement instrument. The enlargement of the covariance matrix R_t for measured data means that we trust this measured data less and more on the prediction. As landmarks are static, we adapt the covariance matrix R_t when the landmark position is updated. Hence, the algorithm for tunning the measurement noise covariance R_t can be derived. In this case, IAE algorithm to adapt the measurement noise covariance matrix R_t is derived. Here the technique known as covariance matching is used. The basic idea of this technique is to make the actual value of the covariance of the residual to be consistent with its theoretical value. The innovation sequence $r_t = (z_t - \hat{z}_{n_t,t})$ has a theoretical covariance as:

$$S_{t,EKF}^{[m]} = G_{\theta_{n_t}} \Sigma_{n_t,t-1}^{[m]} G_{\theta_{n_t}}^T + R_t \tag{20}$$

The actual residual covariance $\hat{C}_t$ can be approximated by its sample covariance, through averaging inside a moving window of size N as follows:

$$\hat{C}_t = \frac{1}{N} \sum_{i=k-N+1}^{t} (r_i^T r_i) \tag{21}$$

If the actual value of covariance $\hat{C}_t$ has discrepancies with its theoretical value, then the diagonal elements of R_t based on the size of this discrepancy can be adjusted. The objective of these adjustments is to correct this mismatch as far as possible. The size of the mentioned discrepancy is given by a variable called the degree of mismatch ($DOM_{t,EKF}$), defined as:

$$DOM_{t,EKF} = S_{t,EKF} - \hat{C}_t \tag{22}$$

The basic idea used to adapt the matrix R_t is as follows: from equation (20) an increment in R_t will increase $S_{t,EKF}$ and vice versa. Thus, R_t can be used to vary

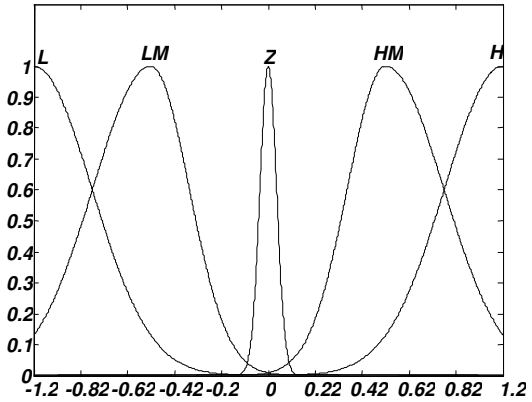


Fig. 1. Inputs Membership functions

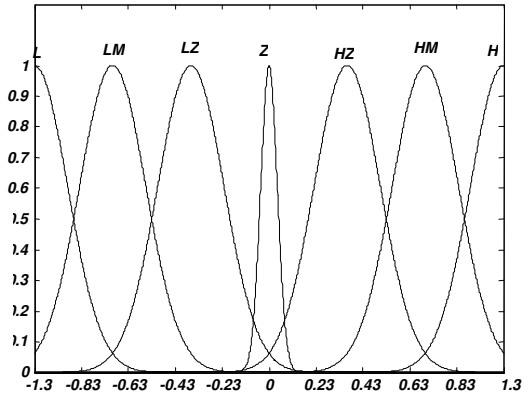


Fig. 2. Output Membership function

$S_{t,EKF}$ in accordance with the value of $DOM_{t,EKF}$ in order to reduce the discrepancies between $S_{t,EKF}$ and $\hat{C}_t$. The adaptation of the (i,i) element of R_t is made according to the (i,i) element of $DOM_{k,EKF}$.

In this paper, ANFIS is proposed to adjust R_t . The overall adaptive Neuro-fuzzy inference system employs a bank of subsystems where each subsystem employs a two-input-single-output ANFIS. This is because the dimensions of DOM_t and R_t are both 2×2 .These two-input-single-output ANFIS are employed to tune each diagonal of element of R_t . The inputs of ANFIS are $DOM_{t,EKF}$ and $DeltaDOM_{t,EKF}$. Here, $DeltaDOM_{t,EKF}$ is defined as:

$$DeltaDOM_{t,EKF} = DOM_{t,EKF} - DOM_{t-1,EKF} \tag{23}$$

Figure 1 presents membership functions for $DOM_{t,EKF}(i,i)$ and $DeltaDOM_{t,EKF}$. Adjustment of R_t is performed using the following equation:

$$R_t = R_t + \Delta R_t \tag{24}$$

Table 1. Rule Table

		Input$_2$				
		L	**ML**	**Z**	**MH**	**H**
Input$_1$	**L**	H	H	HM	HZ	Z
	ML	H	HM	HZ	Z	LZ
	Z	HM	HZ	Z	LZ	LM
	MH	HZ	Z	LZ	LM	L
	H	Z	LZ	LM	L	L

where ΔR_t is the ANFIS output and membership function of output is shown in Fig.2. This ANFIS is a five layers network as shown in Fig.3. The fuzzy rules that complete the ANFIS rule base are as in Table 1.

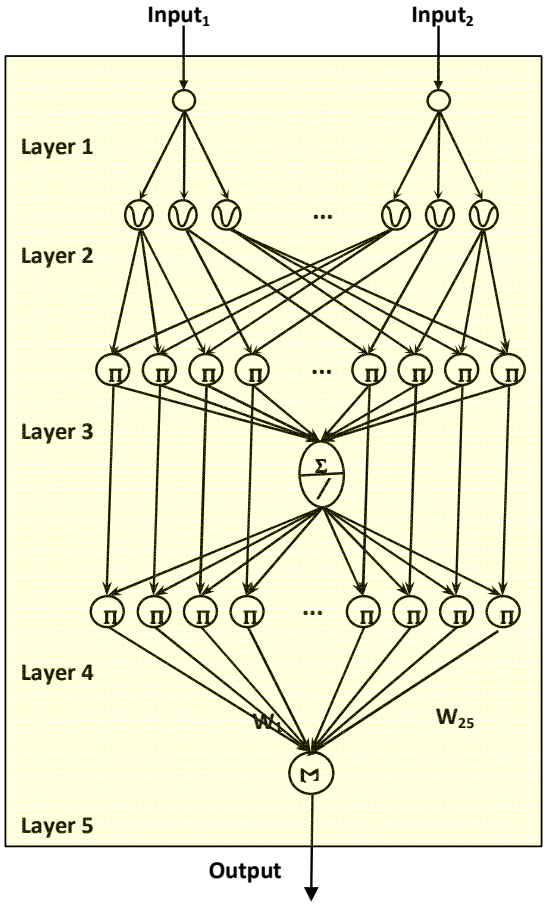


Fig. 3. The Structure of ANFIS for tuning R $_t$ and Q $_t$

The ANFIS is trained using data from the same environment that SLAM will work in it. The training algorithm adjusts the network weights through the minimization of the following cast function:

$$E_t = \frac{1}{2} tr(e_t^2) \tag{25}$$

where

$$e_t = S_{t,EKF} - \hat{C}_t \tag{26}$$

By using the back propagation (BP) learning algorithm, the weighting vector of ANFIS is adjusted such that the error defined in (25) is less than a desired threshold value after a given number of training cycles. The well-known BP algorithm may be written as:

$$W_{t+1} = W_t - \eta \frac{\partial E_t}{\partial W_t} \tag{27}$$

here η is the learning rate and $W_t = [m_t, \sigma_t, \omega_t]^T$ is the tuning parameters of ANFIS. Where m_t and σ_t , respectively, are the center and width of the Gaussian membership function. Also w_l are the link weights in fifth layer. The gradient of E with respect to an arbitrary weighting vector W_t is as follows:

$$\frac{\partial E_t}{\partial W_t} = -e_t \frac{\partial \Delta R_t}{\partial W_t} \tag{28}$$

By the recursive application of the chain rule, the error term for each layer is first calculated, and then the parameters in the corresponding layers are adjusted.

By the recursive application of the chain rule, the error term for each layer is first calculated, and then the parameters in the corresponding layers are adjusted as follows:

$$\begin{aligned} m_{t+1} &= m_t - \eta \tfrac{\partial E}{\partial m_t} \\ \sigma_{t+1} &= \sigma_t - \eta \tfrac{\partial E}{\partial \sigma_t} \\ \omega_{t+1} &= \omega_t - \eta \tfrac{\partial E}{\partial \omega_t} \end{aligned} \tag{29}$$

b) Tuning Q_t

As the covariance matrix R_t is tuned in EKF by first ANFIS in landmark estimation step, it is necessary that only the noise covariance matrix Q_t is tuned adaptively in EKPF in sampling step. The covariance matrix Q_t represents the uncertainty in the process model (odometery). An increase in the covariance matrix Q_t means that we trust less the process model and vice versa. The tuning of Q_t is adaptively adjusted using second ANFIS. The idea behind the process of adaptation Q_t is similar to the adaptation of R_t in EKF. For this purpose, the residual of EKPF using (20) can be rewritten as follows:

$$\begin{aligned} S^{[m]}_{t,EKPF} &= H_t(\nabla f_t P^{[m]}_t \nabla f_t^T + G_u Q_t G_u^T) H_t^T \\ &+ G_{\theta_{n_t}} \Sigma^{[m]}_{n_t,t-1} G^T_{\theta_{n_t}} + R_t \end{aligned} \tag{30}$$

The actual residual covariance in EKPF is $\hat{C}_t$, similar to the actual residual covariance in (21). It may be deduced from equation (29) that a variation in Q_t will affect the value of $S_{t,EKPF}$. If the covariance matrix Q_t is increased then $S_{t,EKPF}$ is increased and vice versa. Thus,

if a mismatch between $S_{t,EKPF}$ and $\hat{C}_t$ is observed then a correction can be made through augmenting or diminishing the value of Q_t . The three general adaptation rules are defined as follows:

1. If $DOM_{t,EKPF}(1,1)$ is Low and $DOM_{t,EKPF}(2,2)$ is Low then ΔQ_t is High.

2. If $DOM_{t,EKPF}(1,1)$ is Zero and $DOM_{t,EKPF}(2,2)$ is Zero then ΔQ_t is Zero.

3. If $DOM_{t,EKPF}(1,1)$ is High and $DOM_{t,EKPF}(2,2)$ is High then ΔQ_t is Low.

The covariance matrix Q_t is adapted as follows:

$$Q_t = Q_t \Delta Q_t \tag{31}$$

where ΔQ_t is the ANFIS output and $DOM_{t,EKPF}(1,1)$ and $DOM_{t,EKPF}(2,2)$ are ANFIS inputs. The second ANFIS model is similar to first ANFIS used for tuning R_t and is a two-input-single-output Neuro-Fuzzy inference system. However, its input is different. The inputs of second ANFIS are $DOM_{t,EKPF}(1,1)$ and $DOM_{t,EKPF}(2,2)$. The structure of second ANFIS is as Fig.3.

3.2 Modifying of Sampling using PSO

To merge PSO into particle filter, we must define a fitness function. The fitness function must consider the newest observations and in addition the fitness function particles with high likelihood should have small values. For this purpose, we propose a fitness function as follows:

$$\tilde{f}_0(s_t) = (z_t - \hat{z}_{n_t,t})^T \left[S_t^{[m]}\right]^{-1} (z_t - \hat{z}_{n_t,t}) \tag{32}$$

where $S_t^{[m]}$ is the residual covariance matrix defined in (11), $\hat{z}_{n,t}$ is the predicted measurement and z_t is the actual measurement. The particles should be moved such that the fitness function is optimal. This is done by tuning the position and velocity of the PSO algorithm. The standard PSO algorithm has some parameters that need to be specified before use. Most approaches use uniform probability distribution to generate random numbers. However it is difficult to obtain fine tuning of the solution and escape from local minima using a uniform distribution. Hence, we use velocity updates based on the Gaussian distribution. In this situation, there is no more need to specify the parameter learning factors c_1 and c_2 . Furthermore, using the Gaussian PSO, the inertial factor ω was set to zero and an upper bound for the maximum velocity $v_{\max}$ is not necessary anymore [22]. Therefore, the only parameter to be specified by the user is the number of particles. Initial values of the particle filter are selected as the initial population of PSO. Initial velocities of PSO are set equal to zero. The PSO algorithm updates the velocity and position of each particle by the following equations [22]:

$$s_t^{[i]} = s_{t-1}^{[i]} + \vec{v}_t^{[i]} \tag{33}$$

$$\vec{v}_t^{[i]} = |randn| \left(P_{pbest}^{[i]} - s_{t-1}^{[i]}\right) + |randn| \left(P_{gbest} - s_{t-1}^{[i]}\right) \tag{34}$$

where $P_{pbest}^{[i]}$ denote the best position that particle i has achieved so far, and P_{gbest} the best of $P_{pbest}^{[i]}$ for any $i = 1, ..., N$. The PSO moves all particles towards the high likelihood regions which is the global best of PSO.

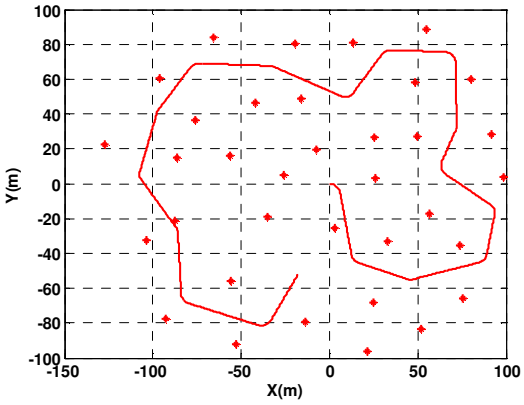


Fig. 4. Experiment environment

When the best fitness value reaches a certain threshold, the optimized sampling process is stopped. With this set of particles the sampling process will be done on the basis of the proposal distribution.

4 RESULTS

4.1 Simulation

Simulation experiments have been carried out to evaluate the performance of the proposed approach in comparison with FastSLAM2.0. The proposed solution for the SLAM problem has been tested for the benchmark environment, with varied number and position of the landmarks, available in [19]. Fig.4 shows the robot trajectory and landmark location. The star points (*) depict the location of the landmarks that are known and stationary in the environment. The initial position of the robot is assumed to be $x_0 = 0$. The robot moves at a speed 3m/s and with a maximum steering angle of 30 degrees. In addition, the robot has 4 meters wheel base and is equipped with a range-bearing sensor with a maximum range of 20m and a 180 ^{o} frontal field-of-view.

The control noise is $\sigma_v = 0.3$ m/s and $\sigma_\gamma = 3^o$. A control frequency is 40 Hz and observation scans are obtained at 5 Hz . The measurement noise is 0.1m in range and

0.1^o in bearing. Data association is assumed unknown. To evaluate the proposed method, its performance is compared with FastSLAM2.0 and improved FastSLAM (IFastSLAM) [16] for the benchmark environment.

First, we consider the situation where measurement noise is wrongly considered as $\sigma_r = 0.01$, $\sigma_\theta = 0.01$. The performance of the proposed method is compared with IFastSLAM and FastSLAM2.0 where its measurement covariance matrix R_t is kept static throughout the experiment with 35 landmarks. The proposed algorithm uses wrongly statistics and then adapts R_t and Q_t matrices in EKPF and EKF through ANFIS and attempts to minimize the mismatch between the theoretical and actual values of the innovation sequence in EKF and EKPF. The free parameters of ANFIS are automatically learned by GD during training.

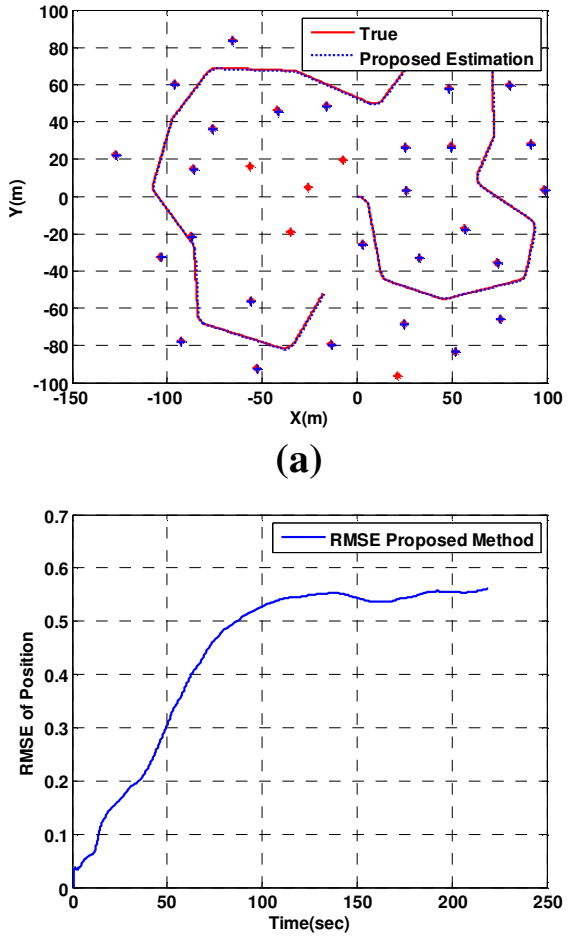


Fig. 5. Proposed method: a) Estimated robot path and estimated landmark with true robot path and true landmark. The "..." is the estimated path, the "+" are the estimated landmark positions. b) Root mean square error (RMSE) of position.

Figure 5 and Fig. 7 show the comparison between algorithms. It can be clearly seen that the results of the proposedalgorithm are better than those of FastSLAM2.0 and IFastSLAM. In other words, in the proposed algorithm, the estimated vehicle path and estimated landmark coincide as close as possible with the actual path and the actual positions landmarks. This is because the proposed method adaptively tuned the measurement covariance matrix R_t and process noise covariance matrices Q_t . In fact, these matrices converges to the actual covariance matrices R_t and Q_t while matrices R_t and Q_t in FastSLAM2.0 are kept fixed over time. Fig.6 and Fig.8 show that measurement covariance matrix R_t converges to the actual covariance matrix R_t in proposed method. In addition, Fig.9 shows results for IFastSLAM. It can be seen that the IFastSLAM is more accurate than FastSLAM2.0 in this situation

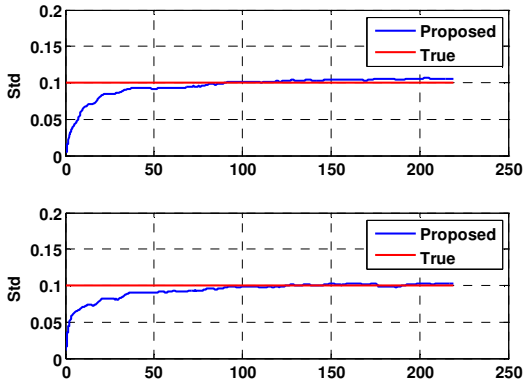


Fig. 6. Measurement covariance matrix of proposed Method.

Finally, we compare the performance of the proposed method and FastSLAM2.0 while the measurement noise and the control noise are similar to the previous experiment and the number of particles decreases in the two algorithms. Fig.10and Fig.11 show results for this case. It is observed that the proposed method is more accurate than FastSLAM2.0 in this situation (such as the previous situation). This is because sample impoverishment improves in the proposed method. The adaptive EKPF is valid when the posterior distribution can be closely approximated by a Gaussian distribution. The proposed method introduced the PSO algorithm into the adaptive EKPF to improve the distribution samples, and speed up the convergence of the particle set. The perturbation particle set which was optimized by the particle swarm makes most of particles that were scattered faraway from true states gather around regions where true states may be present with high probability. Therefore, the effect of each of each particle is enhanced and diversity advantage is improved. Also the impoverishment of particles can be overcome. The performance of the proposed method does not depend on the

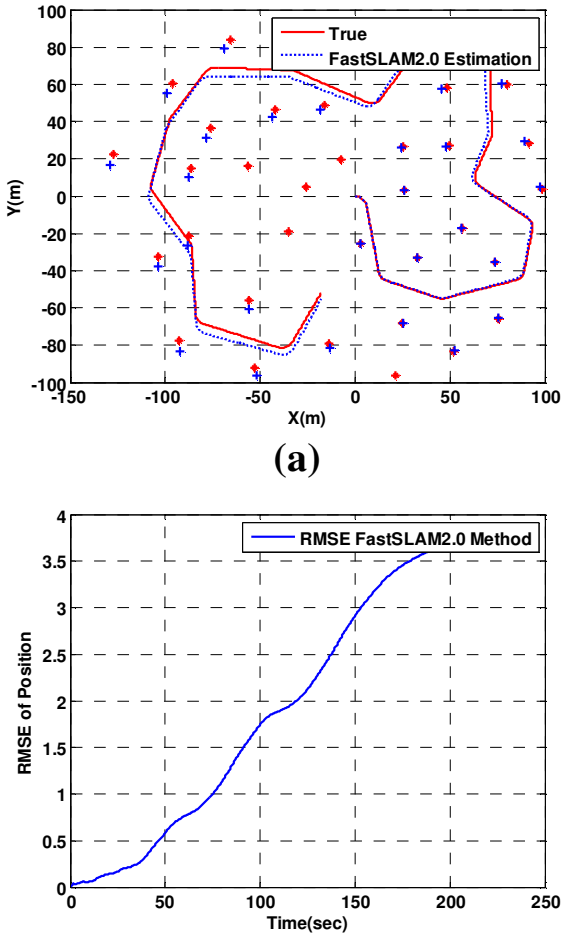


Fig. 7. FastSLAM2.0: a) Estimated robot path and estimated landmark with true robot path and true landmark. The "…" is the estimated path, the "+" are the estimated landmark positions. b) RMSE of position.

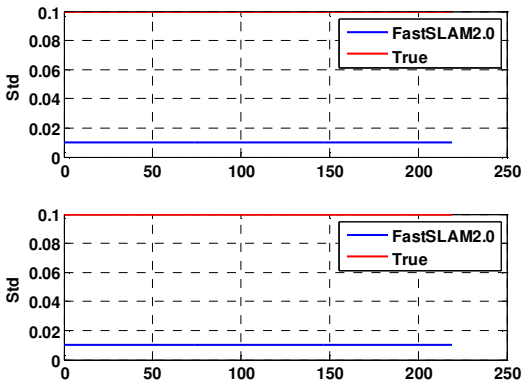


Fig. 8. Measurement covariance matrix of FastSLAM2.0.

number of particles while the performance of FastSLAM

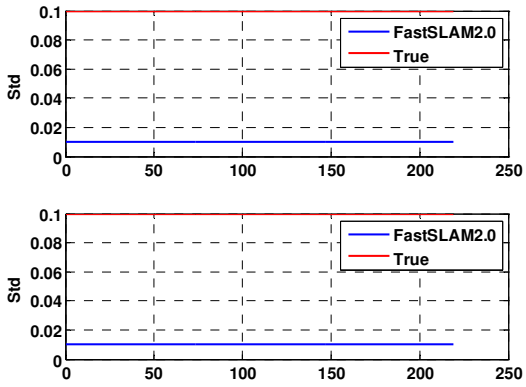


Fig. 9. IFastSLAM: a) Estimated robot path and estimated landmark with true robot path and true landmark. The "…" is the estimated path, the "+" are the estimated landmark positions. b) RMSE of position.

depends on the number of particles. This is because PSO in the proposed method places the particles in the high likelihood region.

4.2 Experimental

The proposed algorithm is compared to FastSLAM2.0 using the car park data set, a popular dataset in the SLAM community. The experimental platform is a 4-wheeled vehicle equipped with a GPS, a laser sensor and wheel encoders as shown in Fig.12.

The vehicle was driven around the park. The velocity and the steering angle were measured with encoders but uneven terrain induced additional non systematic errors because of wheel slippage and vehicle attitude. Consequently, the odometry information from the encoder is poor as shown in Fig. 13. The artificial features were used that consisted of 60 mm steel poles covered with reflective tape. Since the true position of the features was also obtained with GPS, a true navigation map was available for comparison purposes. Also, a kinematic GPS system is used to provide ground truth for the robot position.

The performance of proposed algorithm is compared to the classical FastSLAM2.0 in situation when the correspondences between the observation and the features were assumed unknown and 20 particles were used for both algorithms. Each algorithm was executed many times to confirm the variance of the estimate error. For the unknown data association, the individual compatibility nearest neighbor test is used. The proposed algorithm starts with a wrongly known R_t and then adapts the R_t through ANFIS and attempts to minimize the mismatch between the theoretical and actual values of the innovation sequence. Fig.14 and Fig.15 show the comparison between

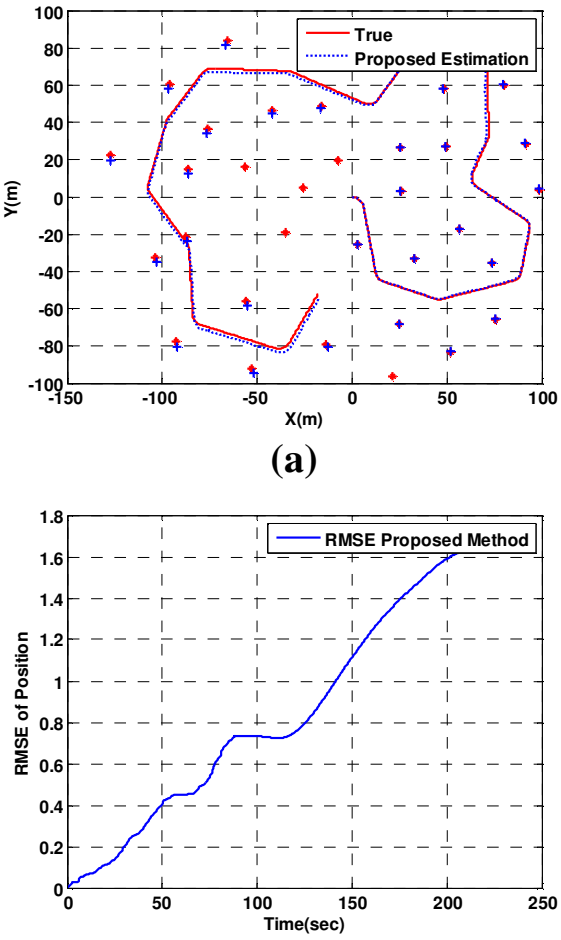


Fig. 10. Proposed method: a) Estimated robot path and estimated landmark with true robot path and true landmark. The "..." is the estimated path, the "+" are the estimated landmark positions. b) RMSE of position.

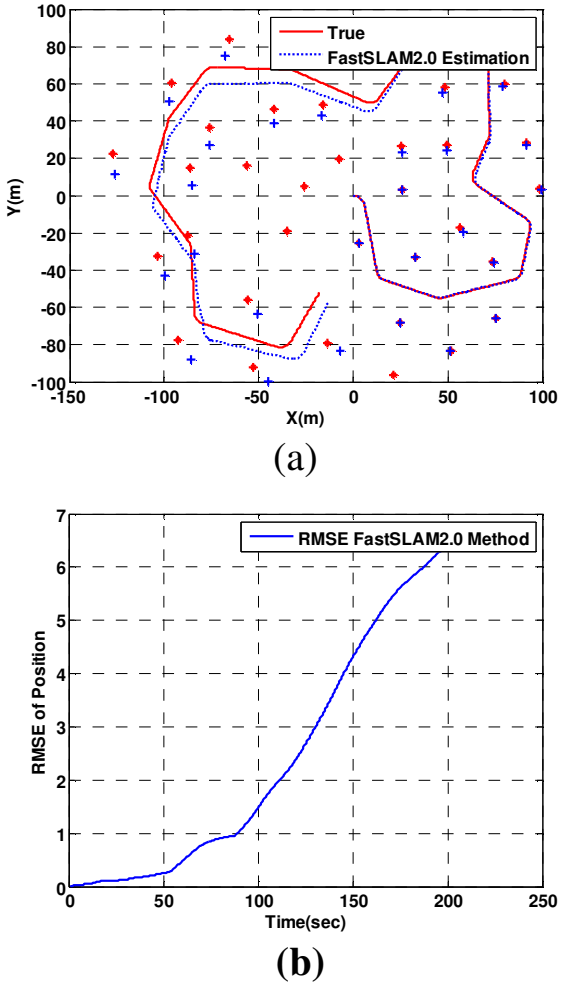


Fig. 11. FastSLAM2.0: a) Estimated robot path and estimated landmark with true robot path and true landmark. The "..." is the estimated path, the "+" are the estimated landmark positions. b) RMSE of position.

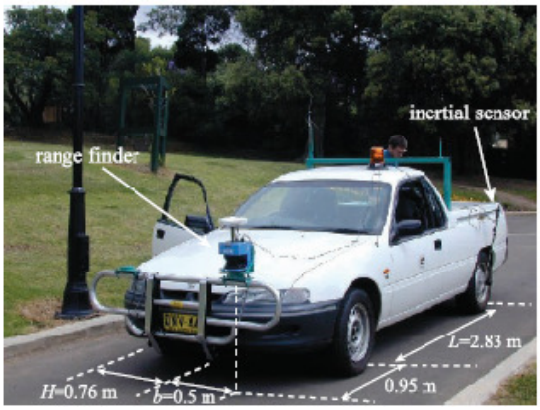


Fig. 12. The vehicle used for experiment.

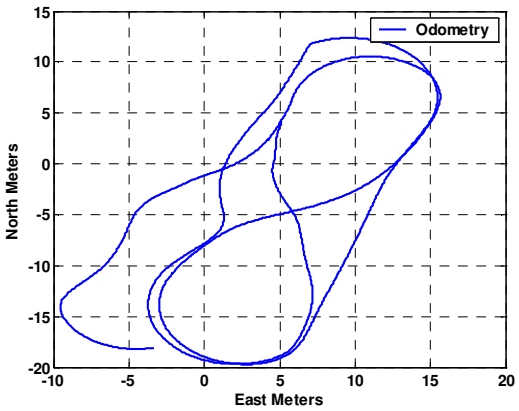
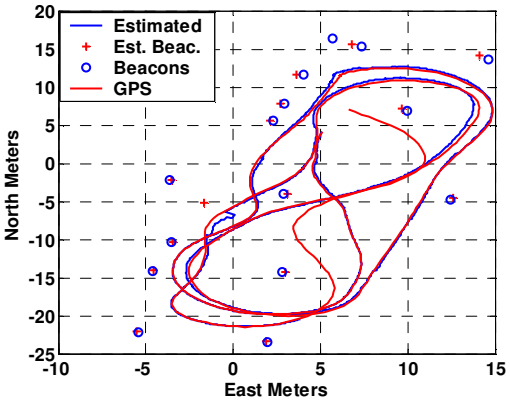
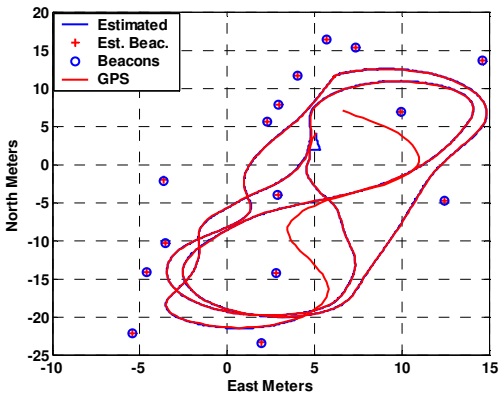


Fig. 13. Odometry of the vehicle.



(a)



(b)

Fig. 14. (a) FastSLAM2.0 (b) Proposed Method. The "..." is the path estimated, the '+' are the estimated beacon positions, the '__' is the GPS path reference and the "o" are the beacon positions given by the GPS.

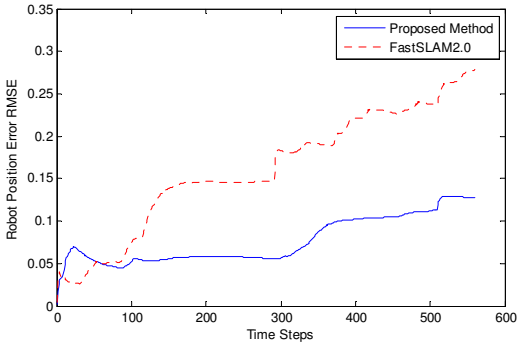


Fig. 15. RMSE of position.

the proposed algorithm and the FastSLAM2.0. Fig.14 shows the trajectory and landmark estimates produced by algorithms, while Fig.15 shows the RMSE of the robot position. The results show that the performance of the proposed algorithm is better than that of FastSLAM2.0. This is because that the proposed algorithm tunes Q_t and R_t adaptively and diversity particles are more than that of FastSLAM2.0. This improves data association, estimation accuracy and consistency.

5 CONCLUSION

This paper presents an intelligent RBPF to solve SLAM problem. In this method, an adaptive EKPF for robot pose estimation, and an adaptive EKF for landmark feature estimation is developed. Then PSO is used to optimize the performance of sampling in the adaptive EKPF. The PSO causes the particle set to tend to the high likelihood region before the weight is updated, thereby the impoverishment of particles can be overcome. Themain advantage of our proposed method is its better accuracy and consistency compared to the classical FastSLAM. This is because in our proposed method, the theoretical value of the innovation sequence matches with its real value.

REFERENCES

[1] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem," IEEE Trans. Robot. Automat. Vol. 17, No. 3, pp.229–241, 2001.

[2] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended Kalman filter based SLAM," IEEE Trans. Robot., vol. 23, no. 5, pp. 1036–1049, Oct. 2007.

[3] M.Montemerlo, S.Thrun, D.Koller, B.Wegbreit, "Fastslam 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in the Int. Joint Conf. on Artificial Intelligence, 2003.

[4] S. Thrun, M. Montemerlo, D. Koller, B.Wegbreit, J.Nieto, E. Nebot, "FastSLAM: An efficient solution to the simultaneous Localization and mapping problem with unknown data association," Journal of Machine Learning Research, 2004.

[5] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," in Proc. IEEE Int. Conf. Robot. Autom., 2003, pp. 1985–1991.

[6] R. Martinez-Cantin and J. A. Castellanos, "Unscented SLAM for largescale outdoor environments," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., 2005, pp. 328–333.

[7] X.Chen, "An Adaptive UKF-Based Particle Filter for Mobil Robot SLAM," International Joint Conference on Artificial Inteligence, 2009.

[8] T.Bailey, J. Nieto and E.Nebot, "Consistency of the FastSLAM algorithm," IEEE Int. Conf, on Robotics and Automation.pp.424-429,

[9] N.Kwak, G.Kim,B.Lee, "A new compensation technique based on analysis of resampling process in FastSLAM," Journal of Robotica, Vol.26, pp. 205–217, 2007.

[10] I.Kim, N.Kwak, H.Lee, B.Lee, "Improved particle fusing geometric relation between particles in FastSLAM," Journal of Robotica, Vol. 27, pp. 853–859, 2009.

[11] X. Wang, H. Zhang, "A UPF-UKF framework for SLAM," IEEE Intl. Conf. on Robotics and Automation, pp 1664-1669, 2007.

[12] M.Li, H.Bing-rong, L.Rong-hua, W.Zhen-Hua, "A novel method for mobile robot simultaneous localization and mapping," Journal of Zhejiang university Scince A, P937-944, 2006.

[13] L.Zhang, X.Meng , Y.Chen, "A FastSLAM Algorithm Based on the Auxiliary Particle Filter with Stirling Interpolation," Proceedings of the 2009 IEEE, International Conference on Information and Automation, June 22 -25, 2009, Zhuhai/Macau, China.

[14] C. Kim, R.Sakthivel, W. K. Chung, "Unscented FastSLAM: A Robust and Efficient Solution to the SLAM Problem," IEEE Trans. Robot., Vol. 24, No. 4, 2008.

[15] R.Havangi, M.A.Nekoui, M., Teshnehlab, "Adaptive Neuro-Fuzzy Extended Kalman Filtering for Robot Localization", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 2, March 2010

[16] R.Havangi, M.A.Nekoui, M., Teshnehlab, "An improved FastSLAM framework using soft computing", Turk J Elec Eng & Comp Sci, Vol.20, No.1, 2012.

[17] R. K. Mehra, On the identification of variances and adaptive Kalman filtering, IEEE Trans Autom Control, Vol. AC-15, No.2, pp. 175–184, Apr, 1970.

[18] R. J. Fitzgerald. "Divergence of the Kalman filter." IEEE Trans. Autom. Control, Vol. AC-16, No. 6, pp. 736–747, Dec. 1971.

[19] R. Krohling, A Gaussian swarm: a novel particle swarm optimization algorithm, In IEEE Conf. on Cybernetics and Intelligent Systems (CIS), Singapore, pp.372-376, 2004.

[20] Kennedy J, Eberhart R C. "Particle Swarm Optimization," Proc. IEEE International Conference on Neural Networks, IV , Piscataway, NJ: IEEE Service Center, 1995. 1942-1948.

[21] M.Clerc, J.Kennedy, "The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space," IEEE Trans On Evolutionary Computation, Vol. 6, No. 1, February 2002.

[22] R. Krohling, "A Gaussian swarm: a novel particle swarm optimization algorithm", IEEE Conference on Cybernetics and Intelligent Systems, Singapore, pp. 372-376, 2004.

R. Havangi received his M.S. and Ph.D. degrees from the K.N. Toosi University of Technology, Tehran, Iran, in 2003 and 2012, respectively. He is currently an Associate Professor of control systems with the Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran. His main research interests are inertial navigation, integrated navigation, estimation and filtering, evolutionary filtering, simultaneous localization and mapping, fuzzy, neural network, and soft computing.

AUTHORS' ADDRESSES

Ramazan Havangi, Ph.D.
University of Birjand,
Faculty of Electrical and Computer Engineering,
South Khorasan Province, Birjand, A78, 97175615, Iran
e-mail: Havangi@Birjand.ac.ir

Received: 2016-01-02
Accepted: 2017-03-20